



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/973,069	10/10/2001	Johan Andersson	66848-0001-2	4674

25269 7590 05/27/2005

DYKEMA GOSSETT PLLC  
FRANKLIN SQUARE, THIRD FLOOR WEST  
1300 I STREET, NW  
WASHINGTON, DC 20005

EXAMINER

PICH, PONNOREAY

ART UNIT	PAPER NUMBER
----------	--------------

2135

DATE MAILED: 05/27/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

**Advisory Action  
Before the Filing of an Appeal Brief**

Application No.

09/973,069

Applicant(s)

ANDERSSON ET AL.

Examiner

Ponnoreay Pich

Art Unit

2135

**--The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**

THE REPLY FILED 11 May 2005 FAILS TO PLACE THIS APPLICATION IN CONDITION FOR ALLOWANCE.

1. ☒ The reply was filed after a final rejection, but prior to or on the same day as filing a Notice of Appeal. To avoid abandonment of this application, applicant must timely file one of the following replies: (1) an amendment, affidavit, or other evidence, which places the application in condition for allowance; (2) a Notice of Appeal (with appeal fee) in compliance with 37 CFR 41.31; or (3) a Request for Continued Examination (RCE) in compliance with 37 CFR 1.114. The reply must be filed within one of the following time periods:

- a) ☒ The period for reply expires 3 months from the mailing date of the final rejection.  
b) ☐ The period for reply expires on: (1) the mailing date of this Advisory Action, or (2) the date set forth in the final rejection, whichever is later. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of the final rejection.  
Examiner Note: If box 1 is checked, check either box (a) or (b). ONLY CHECK BOX (b) WHEN THE FIRST REPLY WAS FILED WITHIN TWO MONTHS OF THE FINAL REJECTION. See MPEP 706.07(f).

Extensions of time may be obtained under 37 CFR 1.136(a). The date on which the petition under 37 CFR 1.136(a) and the appropriate extension fee have been filed is the date for purposes of determining the period of extension and the corresponding amount of the fee. The appropriate extension fee under 37 CFR 1.17(a) is calculated from: (1) the expiration date of the shortened statutory period for reply originally set in the final Office action; or (2) as set forth in (b) above, if checked. Any reply received by the Office later than three months after the mailing date of the final rejection, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**NOTICE OF APPEAL**

2. ☐ The Notice of Appeal was filed on \_\_\_\_\_. A brief in compliance with 37 CFR 41.37 must be filed within two months of the date of filing the Notice of Appeal (37 CFR 41.37(a)), or any extension thereof (37 CFR 41.37(e)), to avoid dismissal of the appeal. Since a Notice of Appeal has been filed, any reply must be filed within the time period set forth in 37 CFR 41.37(a).

**AMENDMENTS**

3. ☐ The proposed amendment(s) filed after a final rejection, but prior to the date of filing a brief, will not be entered because  
(a) ☐ They raise new issues that would require further consideration and/or search (see NOTE below);  
(b) ☐ They raise the issue of new matter (see NOTE below);  
(c) ☐ They are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal; and/or  
(d) ☐ They present additional claims without canceling a corresponding number of finally rejected claims.

NOTE: \_\_\_\_\_. (See 37 CFR 1.116 and 41.33(a)).

4. ☐ The amendments are not in compliance with 37 CFR 1.121. See attached Notice of Non-Compliant Amendment (PTOL-324).  
5. ☐ Applicant's reply has overcome the following rejection(s): \_\_\_\_\_.  
6. ☐ Newly proposed or amended claim(s) \_\_\_\_\_ would be allowable if submitted in a separate, timely filed amendment canceling the non-allowable claim(s).  
7. ☒ For purposes of appeal, the proposed amendment(s): a) ☐ will not be entered, or b) ☒ will be entered and an explanation of how the new or amended claims would be rejected is provided below or appended.  
The status of the claim(s) is (or will be) as follows:  
Claim(s) allowed: \_\_\_\_\_.  
Claim(s) objected to: \_\_\_\_\_.  
Claim(s) rejected: 31-38.  
Claim(s) withdrawn from consideration: \_\_\_\_\_.

**AFFIDAVIT OR OTHER EVIDENCE**

8. ☐ The affidavit or other evidence filed after a final action, but before or on the date of filing a Notice of Appeal will not be entered because applicant failed to provide a showing of good and sufficient reasons why the affidavit or other evidence is necessary and was not earlier presented. See 37 CFR 1.116(e).  
9. ☐ The affidavit or other evidence filed after the date of filing a Notice of Appeal, but prior to the date of filing a brief, will not be entered because the affidavit or other evidence failed to overcome all rejections under appeal and/or appellant fails to provide a showing of good and sufficient reasons why it is necessary and was not earlier presented. See 37 CFR 41.33(d)(1).  
10. ☐ The affidavit or other evidence is entered. An explanation of the status of the claims after entry is below or attached.

**REQUEST FOR RECONSIDERATION/OTHER**

11. ☒ The request for reconsideration has been considered but does NOT place the application in condition for allowance because:  
See attached.  
12. ☐ Note the attached Information Disclosure Statement(s). (PTO/SB/08 or PTO-1449) Paper No(s). \_\_\_\_\_.  
13. ☐ Other: \_\_\_\_\_.

### **DETAILED ACTION**

Claims 1-30 were previously cancelled. Claim 34 was amended. Claims 31-38 are pending.

#### ***Response to Amendment***

The examiner notes that claim 34 was amended.

#### ***Response to Arguments***

Applicant filed an argument dated 5/11/2005 in which applicant made a side-by-side argument that Endicott does not meet the limitations of the claims. Applicant first argues that the objectives as disclosed by Endicott are different from the objective of the applicant's invention. To this, the examiner asserts that the objective is irrelevant because though two inventors might try to solve two different problems, they might arrive at an invention that is also capable of solving the other's problem. It is more important that the fields of the invention be similar. In this case, both Endicott and the applicant's invention deal with programming methodologies. The examiner will now address the arguments in each claim individually.

#### **Claims 31, 36, and 37:**

As per claims 31, 36 and 37, applicant argues that Endicott does not disclose a method, system, or program code for automatic control of real world entities, where the control of an individual entity depends on how it relates to other entities. The examiner notes that before claim 24 was cancelled, claim 24 was directed to controlling real world entities using a computer program. This limitation was substantially similar to the above

Art Unit: 2135

limitation and the applicant did not argue Endicott meeting this limitation when Endicott was first used to make the rejection for claim 24. Further, one can see that the examiner cited passage for the above limitation for claims 31, 36, and 37 refer to a Fig 2. In examining Fig 2, the examiner believes that a John, Joe, Sam, and Steve all represent real world entities. Lawyer, Engineer, Manager, and Engineer II are classes, but they are also real world entities as they represent real world jobs/roles. Note that Joe and John are of type Engineers and there are fields in the Engineer class which could be set from a computer which would control how much salary Joe or John has, where their Development Areas are, and their Current Projects. Note also that Personnel, Personnel II and Finance could also be viewed as real world entities. As we travel down the tree seen in Fig 2, we see that the fields that each real world entity has changes depending on which real world entity was the parent of that entity. It is not unreasonable to assume that the objects and data fields needed to describe a Personnel object would be different from the fields needed to describe a Finance object.

Applicant also argues Endicott does not meet the limitation of creating at least two structures, wherein each structure is based on a certain type of relationship between object instances. Note that in object oriented programming (OOP) every subclass of a class is also inherently an instance of that class. Also, a tree is composed of sub-trees. In the passage cited by the examiner (col 5, lines 42-68) Fig 2 is once again referred to. Fig 2 is clearly a tree structure and as stated, a tree is composed of sub-trees, so in the tree disclosed by Endicott, there is at least two trees which are structures. Each individual object defined in the tree could also be viewed as a

structure. At the root of the tree is an Object class which has children called Personnel, Personnel II, and Finance, and they have children, which also have children. Every child in that tree is inherently also an object instance because of the nature of OOP. Therefore, the above limitation is inherent to the nature of OOP.

Applicant further argues that Endicott does not teach a composite object type, which is an object type that includes at least two other object types. Note in Fig 2, the object, Object, for instance. The examiner asserts that items 202 and 204 are objects which are used to define a name and a class. Object is a class, but it is also an object. In OOP languages such as C++ or Java, composite objects composed of at least two other object types are common; it is part of what allows such great reusability in OOP languages. Applicant also argues that Endicott does not teach the element that at least one of the thus included object types defines functions for the control of a real world entity. Note the Cur. Prj. field in the class/object Engineer. In OOP, it is known that data fields can consist of either primitive data types (such as int or char) or they could be Objects themselves (such as Integer or String). It is not unreasonable to assume that a String object would be used to define what a current project for an engineer is. As the Cur. Prj. field defines the current project the engineer is working on, it is obvious that the Cur. Prj. object defines functions for the control of a real world entity. The examiner also notes that in an earlier office action a rejection of a composite object type limitation was made using Endicott for claim 18, which the applicant cancelled and did not argue. The examiner assumed that the applicant agreed with the rejection that Endicott disclosed a composite object type.

Applicant further argues that Endicott does not teach locating each formal instance in at least one of two groups of formal instances wherein each group is associated with a structure and at least two groups are associated with different structures. The passages cited by the examiner show OOP being used by Endicott. Applicant defined on p9 of the specification that formal instances are data entities that are created to describe the way that objects should be instantiated to obtain real instance of the objects. Formal instances are inherent to OOP, without them, real instances cannot be defined. Note in Fig 2 that each double lined box represents a class, but they also show structures and groups of formal instances and they show what fields need to be defined for a real instance of an object.

As per the limitation of instantiation of the composite object type, wherein for each group of formal instances corresponding real world object instance are created and located in the structure with which the group is associated, applicant argues that Endicott and applicant's objectives are different. As state before, objectives are irrelevant. Applicant states that the cited passage does not meet the applicant's objective, but the examiner asserts that it is the claim limitations that need to be met by Endicott for a valid rejection, not the applicant's objectives.

As per the limitation of the final limitation of claims 31, 36 and 37, applicant argues that the objective of the applicant is different than Endicott, there is no mention of control functions for control of real world entity in Endicott, and no mention of how such control function is adapted to how the entity is related to other entities when the object is placed in the object structures. The first two arguments have been addressed

Art Unit: 2135

before and the response by the examiner would be substantially similar in nature, so the examiner will not repeat what has already been said. As per the final argument, as stated, Endicott uses OOP, so each sub-class is also an instance of a class. A sub-class inherits the control function of the parent class, so in that manner, the control function is adapted to how the entity is related to other entities when the object is placed in the object structures.

**Claim 32:**

Applicant argues that Endicott does not disclose any concept similar to a composite object type with formal instances. This argument is similar to the ones already addressed in claims 31, 36, and 37. See the explanations for those claims above.

**Claim 34:**

As per claim 34, applicant starts out with a statement, which says that the concept of formal instances is related to the concept of composite objects and that Endicott does not disclose or describe any concept similar to the concept of composite object. The examiner has explained above how this statement by applicant is not true. The limitation itself reads, wherein a formal instance comprises a description of how the properties of the corresponding object type are to be changed when a real world instance is created. Note applicant changed "real world" to "actual" in the arguments filed 5/11/2005, but the examiner's arguments still hold for either case.

Note that in one of the passages cited by the examiner (col 12, line 54-col 13, line 54) Fig 8A is referred to. Actual instance John is also referred to. Note that John

Art Unit: 2135

was created from formal instance Engineer (see Fig 2). In Fig 8A, we see that John has each of his fields/properties defined. The examiner submits that each of the fields in the formal instance Engineer reads on a description of how the properties of the corresponding object type are to be changed when the actual instance John is created. Obviously, since John in Fig 8A had each of his properties changed properly to an appropriate field (i.e. Object Name was changed to John, Emp# was changed to 453690, ect.), the field names of Engineer was enough of a description for a user to have changed the properties of John when John was created. Further, note that on col 12, line 62 that John had a function call John.update\_salary. It is not unreasonable to assume that when John was first created, if the user did not immediately set the fields/properties of John during creation, there would exist a function call of equally descriptive nature as update\_salary which could be used to first set John's salary for example. Such an update is usually done immediately after John was created, because most programmers would want the fields of John to be initialized or a run time error could occur if a field was invoked without being initialized. Thus, the examiner believes that applicant's arguments for claim 34 is incorrect.

**Claim 35:**

As per claim 35, applicant argues Endicott does not show how a formal instance can represent a group of formal instance. This argument is substantially similar to ones already addressed in claims 31, 36, and 37. See the explanations for those claims above. Also note that in OOP, as a subclass can be both a formal instance as well as an real/actual instance, a formal instance of a subclass is representative of each of the



class it inherits from and in that way represents a group of formal instances. In Fig 2, the class Engineer is a formal instance, but it is also an actual instance of Personnel and Object, which are both formal instances. Personnel and Object together comprise a group of formal instance. The examiner also notes that this limitation is identical to claim 14 which was cancelled and not argued by the applicant when Endicott was applied in an earlier office action. Therefore, the examiner assumed that applicant agreed that the rejection with Endicott was valid.

**Claim 38:**

As per claim 35, applicant argues that the section cited by the examiner does not show a computer program wherein the program code means are stored in a computer readable medium. The examiner asserts that the passage and corresponding figures cited by the examiner shows that Endicott's invention must be implemented via a computer, therefore a program code means are stored in a computer readable medium is inherent to Endicott. The examiner also notes that this limitation is identical to claim 23 which was cancelled and not argued by the applicant when Endicott was applied in an earlier office action. Therefore, the examiner assumed that applicant agreed that the rejection with Endicott was valid.

**Claim 33:**

As per claim 33, applicant argues that the present invention deals with object instance not object classes. The examiner has stated already that in OPP, a sub-class is inherently an instance of the parent class. See above for a more detailed explanation. Applicant also argues that Endicott does not describe the parts of Dog as

Art Unit: 2135

objects. In other words, applicant is arguing Endicott does not teach composite objects. This argument has been addressed already. Applicant argues Endicott does not mention anything about organizing object instances into structures. In Fig 2, each subclass is inherently an instance of the parent class. The instances being organized into a tree hierarchy shows that applicant's argument of Endicott not teaching organizing object instances into structures to be incorrect. Finally, applicant argues that "eyes" and "legs" are objects located with the object "Dog" is incorrect. Applicant is basically arguing that Endicott does not disclose a structure based on physical location. First, the examiner would like to note that in OOP languages such as Java, any data that is defined using a primitive type could also be defined using an object. Further, as an object is an abstraction of a physical or logical resource, applying broadest reasonable interpretation, even primitive types are technically objects. Finally, eyes and legs are inherently parts of a dog, so a structure which defines the eyes and legs as located on a dog is inherent. A structure based on location can also be more clearly seen in Fig 2. One of the fields of Engineer is Dev. Area, which defines the development area or location of an engineer.

### ***Conclusion***


Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ponnoreay Pich whose telephone number is 571-272-7962. The examiner can normally be reached on 8:00am-4:30pm Mon-Fri.

Art Unit: 2135

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kim Vu can be reached on 571-272-3859. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

PP



KIM VU  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100